```perl
#!/usr/bin/perl
use DBI;
$myurl = 'http://MoneyOrdersOnline.com/a';
&main;

sub main{
    $dbh = DBI->connect('DBI:mysql:moneyordersonline:localhost', 'moo', '',
{RaiseError => 0, PrintError => 0});
    unless ($dbh)   {
        print_file_hinted('techdiff.html', {error => 'cannot access
database'});
        return;    }
    &go;
    $dbh->disconnect;}

sub go{
    my($query);
    $query = parse_query();
    if ($query->{action} eq 'lookup_ebay')   {lookup_ebay($query); return;}
    if ($query->{action} eq 'have_info')     {have_info($query); return;}
    if ($query->{action} eq 'choice')        {choice($query); return;}
    if ($query->{action} eq 'download')         {download($query); return;}
    if ($query->{trans_approved})            {from_mol($query); return;}
    if ($query->{secret})                        {from_email($query);
return;}
    $query->{error} .= "<BR>unknown action [$query->{action}]\n";
    print_file_hinted('techdiff.html', $query);}

sub lookup_ebay{
    my($query) = @_;
    # Here is were we look up the ebay item...
    @$query{qw(error description buyer price seller)} =
retrieve_ebay_info($query->{itemno});
    #@$query{qw(error description buyer price seller)} = ('', 'A nice pretty
widget', 'tim@nerds.com', '33.22', 'brianvon@fpga.com');
    # If the buyer or seller looks like an email address, it probably is...
    @$query{b_email, b_email2} = @$query{buyer, buyer} if $query->{buyer} =~
/^[0-9\-\.\_a-z]+@[0-9\-\.\_a-z]+$/;
    @$query{s_email, s_email2} = @$query{seller, seller} if $query->{seller}
=~ /^[0-9\-\.\_a-z]+@[0-9\-\.\_a-z]+$/;
    print_file_hinted('moreinfo.html', $query);}

sub have_info{
    my($query) = @_;
    my($sth, $n);
    # Validate the information
    my($error);
    for (qw(price shipping
            b_fname b_lname b_address b_city b_state b_zipcode
            s_fname s_lname s_address s_city s_state s_zipcode))
        {        $error .= "<LI>Specify the $_" unless $query->{$_};      }
    $error =~ s/\bb_/buyer /g;
```

```perl
$error =~ s/mname/middle name/g;
$error =~ s/lname/last name/g;
if ($query->{b_email} && $query->{b_email} ne $query->{b_email2})
{          $error .= "<LI>The buyer's email addresses must match";        }
if ($query->{s_email} && $query->{s_email} ne $query->{s_email2})
{          $error .= "<LI>The seller's email addresses must match";        }
if ($error)      {
    $query->{error} = "Correct the following errors:<UL>$error</UL>";
    print_file_hinted('moreinfo.html', $query);
    return;      }
# Calculate fee...
$query->{amount} = $query->{price} + $query->{shipping};
$query->{fee} = sprintf("%0.2f", ($query->{amount} * .04) - 0.01);
$query->{fee} = sprintf("%0.2f", ($query->{fee} < 1.00) ? 0.99 : $query-
>{fee});
$query->{total} = $query->{amount} + $query->{fee};
# Create a new check number
my($checkno);
# Pick a random transaction number and make sure it isn't used...
do{  $checkno = int(rand(8999999)) + 1000000;
    $n = $dbh->do('select NULL from transaction where transnumber = ?',
            undef, $checkno);      }
while ($n >= 1);
$query->{checkno} = $checkno;
$sth = $dbh->prepare("replace into transaction set" .
            " modified = UNIX_TIMESTAMP(), transnumber = ?," .
            " b_email = ?, b_fname = ?, b_mname = ?, b_lname = ?,
b_address = ?, b_city = ?, b_state = ?, b_zipcode = ?," .
            " s_email = ?, s_fname = ?, s_mname = ?, s_lname = ?,
s_address = ?, s_city = ?, s_state = ?, s_zipcode = ?," .
            " description = ?, itemno = ?, notify = ?, price = 100 *
?, shipping = 100 * ?, fee = 100 * ?," .
            " b_secret = ?, s_secret = ?");
$n = $sth->execute(@$query{qw(checkno
            b_email b_fname b_mname b_lname b_address b_city b_state
b_zipcode
            s_email s_fname s_mname s_lname s_address s_city s_state
s_zipcode
            description itemno notify price shipping fee b_secret
s_secret)});
if ($n != 1)      {
    $query->{error} .= "<BR>in have_info: select transaction n=[$n]\n";
    print_file_hinted('techdiff.html', $query);
    return;      }
$query->{image} = "http://moneyordersonline.com/check/check.cgi/.gif"
                    . "?checkno=" . URL::encode($query->{checkno})
                    . "&itemno=" . URL::encode($query->{itemno})
                    . "&payee=" . URL::encode("@$query{s_fname,
s_lname}")
                    . "&amount=" . URL::encode($query->{amount});
print_file_hinted('howpay.html', $query);}
```

```perl
    my $checkno = $query->{'checkno'} || $query->{'user2'};
    my $itemno = $query->{'itemno'} || $query->{'user3'};
    unless ($checkno)    {
        $query->{error} .= "<BR>check number is missing\n";
        print_file_hinted('techdiff.html', $query);
        return;     }
    if ($query->{trans_approved} ne 'APPROVED')    {
        $query->{error} .= "<BR>transaction was not approved\n";
        print_file_hinted('techdiff.html', $query);
        return;     }
    my($sth, $n);
    $sth = $dbh->prepare("select" .
        " b_email, b_fname, b_mname, b_lname, b_address, b_city, b_state,
b_zipcode," .
        " s_email, s_fname, s_mname, s_lname, s_address, s_city, s_state,
s_zipcode," .
        " description, itemno, notify, price/100, shipping/100, fee/100," .
        " b_secret, s_secret" .
        " from transaction where transnumber = ?");
    unless ($sth)    {
        $query->{error} .= "<BR>in from_mol: prepare select transaction
failed: @{[DBI->errstr]}\n";
        print_file_hinted('techdiff.html', $query);
        return;     }
    $n = $sth->execute($checkno);
    if ($n != 1)
    {    $query->{error} .= "<BR>select transaction n=[$n]\n";
        print_file_hinted('techdiff.html', $query);
        return;     }
    my($data);
    @$data{qw(        b_email b_fname b_mname b_lname b_address b_city b_state
b_zipcode
                    s_email s_fname s_mname s_lname s_address s_city s_state
s_zipcode
                    description itemno notify price shipping fee
                    b_secret s_secret
        )} = $sth->fetchrow_array;
    $sth->finish;
    if ($data->{itemno} != $itemno)
    {    $query->{error} .= "<BR>wrong itemno\n";
        print_file_hinted('techdiff.html', $query);
        return;     }
    $data->{amount} = $data->{price} + $data->{shipping};
    $data->{total} = $data->{amount} + $data->{fee};
    if ($data->{total} != $query->{trans_amount})
    {    $query->{error} .= "<BR>in from_mol: transaction amount is wrong
($data->{total} vs. $query->{trans_amount}\n";
        print_file_hinted('techdiff.html', $query);
        return;     }
    $data->{b_email} = $query->{email};
    $data->{b_fname} = $query->{firstname};
    $data->{b_lname} = $query->{lastname};
```

```
        $data->{b_city} = $query->{city};
        $data->{b_state} = $query->{state};
        $data->{b_zipcode} = $query->{zip};
        $data->{status} = 'PAID CC';
        $data->{cc_type} = $query->{cardtype};
        $data->{cc_number} = $query->{card_number};
        $data->{cc_exp} = "$query->{exp_mon}/$query->{exp_year}";
        $data->{b_secret} = int(rand(89999) + 10000);
        $data->{s_secret} = int(rand(89999) + 10000);
        $data->{checkno} = $checkno;
        if (0)
        {       &print_header;
                print "<P>in from_mol: data:\n";
                for (sort keys %$data)
                {       print "<BR>   $_ -> $data->{$_}\n";            }      }
        $sth = $dbh->prepare("update transaction set" .
                    " modified = UNIX_TIMESTAMP()," .
                    " b_email = ?, b_fname = ?, b_mname = ?, b_lname = ?,
b_address = ?, b_city = ?, b_state = ?, b_zipcode = ?," .
                    " status = ?, cc_type = ?, cc_number = ?, cc_exp = ?," .
                    " b_secret = ?, s_secret = ? where transnumber = ?");
        unless ($sth)
        {       $query->{error} .= "<BR>in from_mol: prepare update transaction
failed: @{[DBI->errstr]}\n";
                print_file_hinted('techdiff.html', $query);
                return;      }
        $n = $sth->execute(@$data{qw(        b_email b_fname b_mname b_lname
b_address b_city b_state b_zipcode
                                                status cc_type cc_number
cc_exp
                                                b_secret s_secret checkno
                                     )}));
    if ($n != 1)
    {       $query->{error} .= "<BR>in from_mol: execute failed n=[$n]\n";
            print_file_hinted('techdiff.html', $query);
            return;      }
    my($m, $n);
    if ($data->{notify} eq 'yes') {
    $m = <<'EOF';
From: MoneyOrdersOnline.com <Fulfillment@MoneyOrdersOnline.com>
To: {b_email}
Subject: Money Order for eBay Item #{itemno}

To Money Orders Online.com Customer:

Your purchase of a money order in the amount of ${amount} is complete.
The money order has been delivered to the seller {seller}.  Including
the processing fee of ${fee}, a total amount of ${total} has been
charged to your {cc_type} credit card {cc_number}.

If for some reason you wish to cancel this money order, and the seller
has not yet retrieved the money order, you can go to
```

to cancel.

Thank you for using MoneyOrdersOnline.com!
EOF

```
      $n = <<'EOF';
From: MoneyOrdersOnline.com <Fulfillment@MoneyOrdersOnline.com>
To: {s_email}
Subject: Money Order for eBay Item #{itemno}
```

We have been authorized to send you a money order on behalf of eBay
user {buyer} for eBay item number #{itemno}. The amount of the money
order is ${amount}. You can go to the following website and print the
money order instantly on your own PC-compatible printer on plain paper
and cash it just the same way as any other money order.

{{myurl}}/process.cgi?itemno={itemno}&checkno={checkno}&secret={s_secret}

If you do not have a PC-compatible printer, go to the same site and request
the money order be sent to you via first-class mail.

By the way, you can become a Money Orders Online PREFFERRED SELLER and get
free money order stock and address labels just for letting people know that
you accept money orders online as payment for your auctioned items. Go to
http://MoneyOrdersOnline.com/preferred/intro.html
for more information. Being a preferred seller makes shipping your items a
snap, with automatically printed address labels, official money order
stock, and even preprinted postage with adhesive backing! Streamline your
logistics and shipping operations by joining today!
EOF

```
      } else {
          $m = <<'EOF';
From: MoneyOrdersOnline.com <Fulfillment@MoneyOrdersOnline.com>
To: {b_email}
Subject: Money Order for eBay Item #{itemno}
```

To Money Orders Online.com Customer:

Your purchase of a money order in the amount of ${amount} is complete.
Including the processing fee of ${fee}, a total amount of ${total} has
been charged to your {cc_type} credit card {cc_number}.

You have requested local delivery of your money order. Please go to
the following URL to retrieve and print your money order. Please note
that the money order can be printed only on PC-compatible computers.

{{myurl}}/process.cgi?itemno={itemno}&checkno={checkno}&secret={b_secret}

Thank you for using MoneyOrdersOnline.com!
EOF

```
      }
```

```
}

sub from_email{
     my($query) = @_;
     my $data = lookup([qw(status b_fname b_lname b_secret s_fname s_lname
s_secret itemno price shipping fee)], $query, 'download');
     return unless $data;
     @$data{checkno, itemno} = @$query{checkno, itemno};
     if ($data->{status} =~ /^PRINTED\b/)
     {     $query->{error} .= "<BR>Money order was already $query{status}\n";
           print_file_hinted('techdiff.html', $query);
           return;     }
     my $who = $data->{who};
     unless ($who)
     {     $query->{error} .= "<BR>in from_email: Logic error: no who\n";
           print_file_hinted('techdiff.html', $query);
           return;     }
     $fn = 'click-buyer.html' if $who eq 'buyer';
     $fn = 'click-seller.html' if $who eq 'seller';
     $data->{image} = "http://moneyordersonline.com/check/check.cgi/.gif"
                               . "?checkno=" . URL::encode($data->{checkno})
                               . "&itemno=" . URL::encode($data->{itemno})
                               . "&payee=" . URL::encode("@$data{s_fname,
s_lname}")
                               . "&amount=" . URL::encode($data->{amount});
     print_file_hinted($fn, $data);}

sub choice{
     my($query) = @_;
     my $data = lookup([qw(status s_fname s_lname price shipping fee b_secret
s_secret)], $query, 'choice');
     return unless $data;
     @$data{disp, secret} = @$query{disp, secret};
     if (0)
     {     &print_header;
           print "status=$data->{status}  secret=$data->{secret}\n";
           return;     }
           if ($data->{status} =~ /^PRINTED\b/)
     {     $query->{error} .= "<BR>Money order was already $query{status}\n";
           print_file_hinted('techdiff.html', $query);
           return;     }
     if ($data->{who} eq 'buyer' && $data->{disp} eq 'p-b-mail'
           || $data->{who} eq 'seller' && $data->{disp} eq 'p-s-mail')
     {     print "Content-type: text/html\n\ncheck will be mailed to $data-
>{who}\n";
           return;     }
     if ($data->{who} eq 'seller' && $data->{disp} eq 'p-s-decline')
     {     print "Content-type: text/html\n\nBuyer will be notified that you
declined";
           return;     }
     print_file_hinted('download.html', $data);}
```

```perl
    my $data = lookup([qw(b_fname b_lname b_secret s_fname s_lname s_secret
itemno price shipping fee)], $query, 'download');
    return unless $data;
    my $who = $data->{who};
    unless ($who)
    {    $query->{error} .= "<BR>in download: Logic error: no who
secret=$query->{secret}\n";
         print_file_hinted('techdiff.html', $query);
         return;    }
    if ($who eq 'buyer' && $query->{disp} eq 'p-b-mail'
        || $who eq 'seller' && $query->{disp} eq 'p-s-print'        )
    {    $payee = "@$data{qw(s_fname s_lname)}"; }
    elsif ($who eq 'buyer' && $query->{disp} eq 'p-b-deposit')
    {    $payee = "@$data{qw(b_fname b_lname)}"; }
    else {$query->{error} .= "<BR>in download: who=$who  disp=$query->{disp}
payee=$payee\n";
         print_file_hinted('techdiff.html', $query);
         return;    }
    my $memo = "eBay item #$data->{itemno}";
    my $echeck = load_file('echeck.nif');
    my $download = $echeck . join(';', today(), $payee, $data->{amount},
$memo, $data->{checkno}, '', $secret);
    print "Content-type: application/x-nicotine\nContent-length:
@{[length($download)]}\n\n", $download;}

sub lookup{
    my($fields, $query, $where) = @_;
    my($f);
    for (@$fields)
    {    $f .= ', ' if $f;
         $f .= "$_";
         $f .= '/100' if /^(price|shipping|fee)$/;     }
    if (0)
    {    &print_header;
         print "<BR>fields=@$fields\n";
         print "<BR>select $f from transaction where transnumber = ? and
itemno = ?\n";
         return;    }
    my $sth = $dbh->prepare("select $f from transaction where transnumber =
? and itemno = ?");
    unless ($sth)
    {    $query->{error} .= "<BR>in $where: prepare select transaction
failed: @{[DBI->errstr]}\n";
         print_file_hinted('techdiff.html', $query);
         return;    }
    my $n = $sth->execute(@$query{checkno, itemno});
    if ($n != 1)
    {    $query->{error} .= "<BR>in $where: select transaction n=[$n]
[@$query{checkno, itemno}]\n";
         print_file_hinted('techdiff.html', $query);
         return;    }
    my($data);
```

```perl
    @$data{checkno, itemno} = @$query{checkno, itemno};
    $data->{amount} = sprintf("%.2f", $data->{price} + $data->{shipping}) if
exists $data->{price} && exists $data->{shipping};
    $data->{total} = sprintf("%.2f", $data->{amount} + $data->{fee}) if
exists $data->{amount} && exists $data->{fee};
    $data->{who} = 'buyer' if $query->{'secret'} eq $data->{'b_secret'};
    $data->{who} = 'seller' if $query->{'secret'} eq $data->{'s_secret'};
    $data;}

sub isOK{
    return $_[0] =~
/^timkay\@|^brianvon\@|\@moneyordersonline.com$|\@nerds.com$/i;}

sub sendmail{
    my($message) = @_;
    my($to) = $message =~ /^To:\s+(.*?)\n/im;
    local(*S);
    unless (isOK($to) && open(S, "|/usr/lib/sendmail -t"))
    {    print "<BR>E-mail to $to failed\n";
         return 1;    }
    print S $message;
    close(S);
    &print_header; print "<BR>Mail sent to $to\n";}

sub retrieve_ebay_info{
    my($item) = @_;
    my($html, $error, $description, $buyer, $left, $price, $quantity);
    $item =~ tr/0-9//cd;
    return ('Enter a vaild item number.') unless $item =~ /^(\d{8,10})$/;
    print "<BR>get $item...http://cgi.ebay.com/aw-
cgi/eBayISAPI.dll?MfcISAPICommand=ViewItem&item=" . URL::encode($item) .
"\n";
    $html = URL::get('http://cgi.ebay.com/aw-
cgi/eBayISAPI.dll?MfcISAPICommand=ViewItem&item=' . URL::encode($item));
    #print "<BR>HTML:[[[$html]]]\n";
    ($description) = $html =~ /bgcolor="#99CCCC">.*?<b>(.*?)<\/b>/is;
    $error = "Item #$item not found." unless $error || $description;
    ($left) = $html =~ /<font.*?>Time
left<\/font><\/td>.*?<b><font.*?>(.*?)<\/font><\/b>/is;
    $error = 'The auction has not ended.' unless $error || $left =~ /auction
has ended/is;
    ($quantity) = $html =~
/<font.*?>Quantity<\/font><\/td>.*?<b>(.*?)<\/b>/is;
    if ($quantity > 1)
    {    ($price) = $html =~
/<font.*?>Lowest<\/font><\/td>.*?<b>(.*?)<\/b>/is;
         if ($price)         {    $price =~ s/^\$//;    }
         else
         {    $error = 'Item did not sell' unless $error;
              ($price) = $html =~ /<font.*?>Starts
at<\/font><\/td>.*?<b>(.*?)<\/b>/is;}
         unless ($error)
```

```
                ($seller) = $html =~
/<font.*?>Seller<\/font><\/td>.*?<b><a.*?>(.*?)<\/a><\/b>/is;          }
        } else {
            ($price) = $html =~
/<font.*?>Currently<\/font><\/td>.*?<b>(.*?)<\/b>/is;
            if ($price)
            {           $price =~ s/^\$//;
            }           else
            {       $error = 'Item did not sell' unless $error;
                ($price) = $html =~ /<font.*?>Starts
at<\/font><\/td>.*?<b>(.*?)<\/b>/is;     }
            unless ($error)
            {               ($buyer) = $html =~ /<font.*?>High
Bid<\/font><\/td>.*?<b><a.*?>(.*?)<\/a><\/b>/is;
                ($seller) = $html =~ /<font.*?>Seller
\(Rating\)<\/font><\/td>.*?<b><a.*?>(.*?)<\/a><\/b>/is;          }        }
        ($error, $description, $buyer, $price, $seller);}

sub print_file_hinted{
    my($fn, $data) = @_;
    &print_header;
    print subst(load_file($fn), $data);}

sub print_header{
    print "Content-type: text/html\n\n" unless $header_printed;
    $header_printed++;}

sub load_file{
    my($fn) = @_;
    my($s);
    local(*IN);
    if (open(IN, $fn))
    {       read(IN, $s, -s $fn);
            close(IN);       }
    $s || "Technical difficulties\n";}

sub subst{
    my($s, $data) = @_;
    $data->{method} = 'POST';
    $s =~ s/{{(.*?)}}/eval("\$$1")/gie;
    $s =~ s/"{(.*?)}"/quote($data->{$1})/gie;
    $s =~ s/{=(.*?)}/URL::encode($data->{$1})/gie;
    $s =~ s/{(.*?)}/$data->{$1}/gi;
    $s;}

sub quote{
    my($s) = @_;
    $s =~ s/([\\"])/\\$1/g;
    "\"$s\"";}

sub parse_query{
    my($content, $query);
```

```perl
{     my($key,$val) = /^(.+?)=(.*)$/;
      next unless $key;
      $key =~ s/\+/ /g;
      $key =~ s/%([0-9a-z][0-9a-z])/pack(H2,$1)/ige;
      $val =~ s/\+/ /g;
      $val =~ s/%([0-9a-z][0-9a-z])/pack(H2,$1)/ige;
       $val =~ s/^\s|\s$//g;
     $query->{$key} = $val;        }
   if (0)
   {      &print_header;
      print "<P>In parse_query: query:\n";
      for (sort keys %$query)
      {print "<BR>   $_ --> $query->{$_}\n";   }       }
    $query;}

sub today{
    my(@mon) = qw(January February March April May June July August
September October November December);
    my($sec, $min, $hour, $mday, $mon, $year, $wday, $yday, $isdst) =
localtime(time);
    my $today = sprintf "%s %d, %d", $mon[$mon], $mday, 1900 + $year;
    $today;}

package URL;
use Socket;

sub get{
    my($remote, $path) = $_[0] =~ /^(?:http:\/\/)?(.+?)(\/.*)?[\r\n]*$/;
    my($buf, $ret);
    #print "<BR>urlget [$remote] [$path]\n";
    my($name, $aliases, $addrtype, $length, $remoteaddr) =
gethostbyname($remote);
    #print "<BR>found ", join('.', unpack(C4, $remoteaddr)), "\n";
    socket(S, PF_INET, SOCK_STREAM, 6) or print "<BR>socket: $!\n";
    #print "<BR>connecting to $remote...\n";
     my $sin = sockaddr_in(80, inet_aton($remote));
     connect(S, $sin) or print "<BR>connect: $!\n";
    #print "<BR>Connected to $remote\n";
    select(S); $| = 1; select(STDOUT);
    print S "GET $path\r\n";
    while (sysread(S, $buf, 10240))
     {      $ret .= $buf;       }
    close(S);
    $ret;}

sub encode{
    my($z) = @_;
    $z =~ s/([=\?\&\+;<>])/'%'.unpack(H2,$1)/ge;
    $z =~ tr/ /+/;
    $z;}

### Local Variables: ***
```